

PL/SQL Exception

Disciplina

Database Application Development

Leonardo Uchida

leonardo@fiap.com.br

Referências Bibliográficas

Manuais ou Materiais oficiais da Oracle ou Oracle Press

*Oracle9i / 10g PL/SQL * Guia de Consulta Rápida – Celso Henrique Poderoso*

Material criado originalmente por

Salvio Padlipskas

Este material é apenas um guia de estudo e não substitui a leitura da referência bibliográfica e a consulta de anotações de sala de aula

Tratamento de Erros: Exception

- *Após o término desta lição, você estará habilitado para :*
 - *Definir exceções PL/SQL*
 - *Reconhecer exceções não tratáveis*
 - *Utilizar diferentes tipos de manuseios de exceções PL/SQL*
 - *Capturar erros inesperados*
 - *Descrever o efeito da propagação de exceção em blocos aninhados*
 - *Personalizar mensagens de exceções PL/SQL*

Tratamento de Erros: Exception

- *Exceções são todos os erros ou imprevistos que possam ocorrer durante a execução de um bloco PL/SQL.*
- *Quando ocorre um erro, o SGBDR Oracle abandona a área de comandos(aborta a execução) e procura em uma área de exceções(EXCEPTION), o tratamento para a falha.*
- *Erros inesperados ocorridos em tempo real de execução são geralmente decorrentes de uma falha na codificação do programa fonte, falha de hardware, leitura inconsistente de um determinado conteúdo entre outras origens*
- *Como não é possível antecipar a todos os erros, o planejamento para capturá-los são imprescindíveis em aplicações críticas onde não se permite interrupções*

Tratamento de Erros: Exception

- *EM PL/SQL, uma condição de erro é chamada de exception (exceção)*
- *Uma exception pode ser internamente definida ou definida por uma determinada regra na fase de concepção do código fonte*
- *Como se realiza o tratamento da exceção ?*
 - *Através de comandos implementados no código PL/SQL*

Recuperando os códigos de erro Oracle

- *Para recuperar qualquer código de erro e sua respectiva mensagem relativo no acesso ao banco de dados Oracle, pode-se usar as funções internas SQLCODE e SQLERRM*
- *SQLCODE – Retorna o código do erro Oracle*
- *SQLERRM – Retorna o código e mensagem de erro Oracle*
- *O tamanho máximo da mensagem a ser exibida é de 512 caracteres incluindo o código de erro Oracle*
- *Se não ocorrer erro a mensagem exibida é: **ORA-0000: normal, successful completion.***

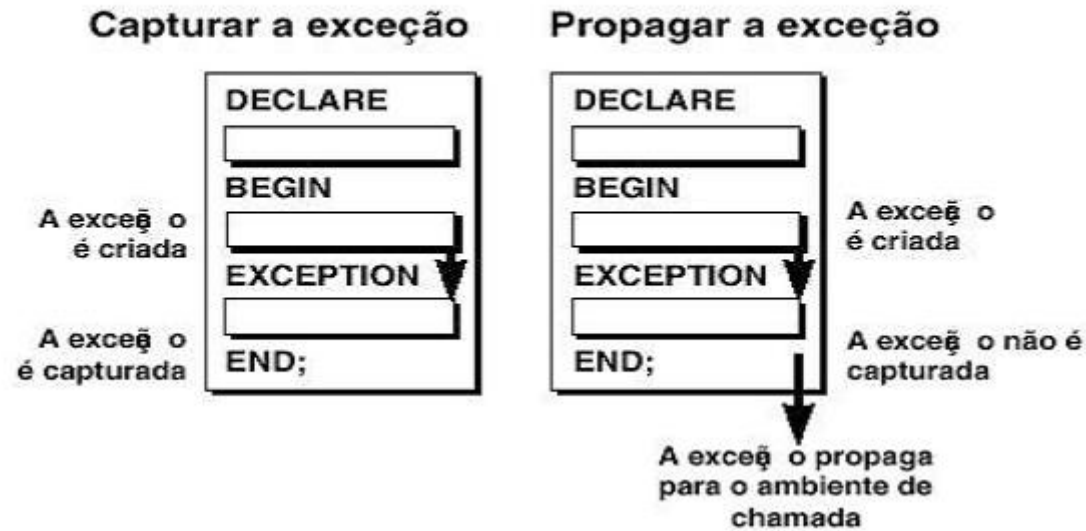
Estrutura de Bloco PL/SQL

- **DECLARE** – Opcional
Variáveis, cursores, exceções definidas pelo usuário
- **BEGIN** – Obrigatório
 - Instruções SQL
 - Instruções PL/SQL
- **EXCEPTION** – Opcional
Ações a serem desempenhadas quando ocorrem erros
- **END;** – Obrigatório

```
DECLARE  
    ○ ○ ○  
BEGIN  
    ○ ○ ○  
EXCEPTION  
    ○ ○ ○  
END;
```

Seção	Descrição	Inclusão
Declarativa	Contém todas as variáveis, constantes, cursores e exceções definidas pelo usuário que são referenciadas nas seções executável e declarativa	Opcional
Executável	Contém instruções SQL para manipular dados no banco de dados e instruções PL/SQL para manipular dados no bloco	Obrigatória
Tratamento de exceção	Especifica as ações a desempenhar quando erros e condições anormais surgem na seção executável	Opcional

Tratando Exceções



Tratamento de Erros: Exception

- *EM PL/SQL, uma condição de erro é chamada de exception (exceção)*
- *Uma exception pode ser internamente definida ou definida por uma determinada regra na fase de concepção do código fonte*
- *Ex: WHEN_NO_DATA_FOUND é uma exceção pré-definida em PL/SQL internamente. CLIENTE_SEM_CREDITO pode ser uma exceção definida pela regra de negócio do projeto CARLOCA*

Tratamento de Erros: Exception

- *Quando um erro ocorre uma exceção é acionada, isto é, a execução normal é interrompida e o controle é transferido para a área de tratamento de exceções, que é uma parte de seu bloco PL/SQL*
- *A chamada de uma exceção implícita é feita automaticamente pelo PL/SQL. Já a chamada de uma exceção definida é chamada pelo comando RAISE*
- *Exemplo de chamada de uma exceção definida pela regra de negócio:
RAISE CLIENTE_SEM_CREDITO*

Tratamento de Erros: Exception

- *Exceções pré-definidas pelo Oracle*
- *Definidas pelo usuário*

Exceção Predefinida

Sintaxe

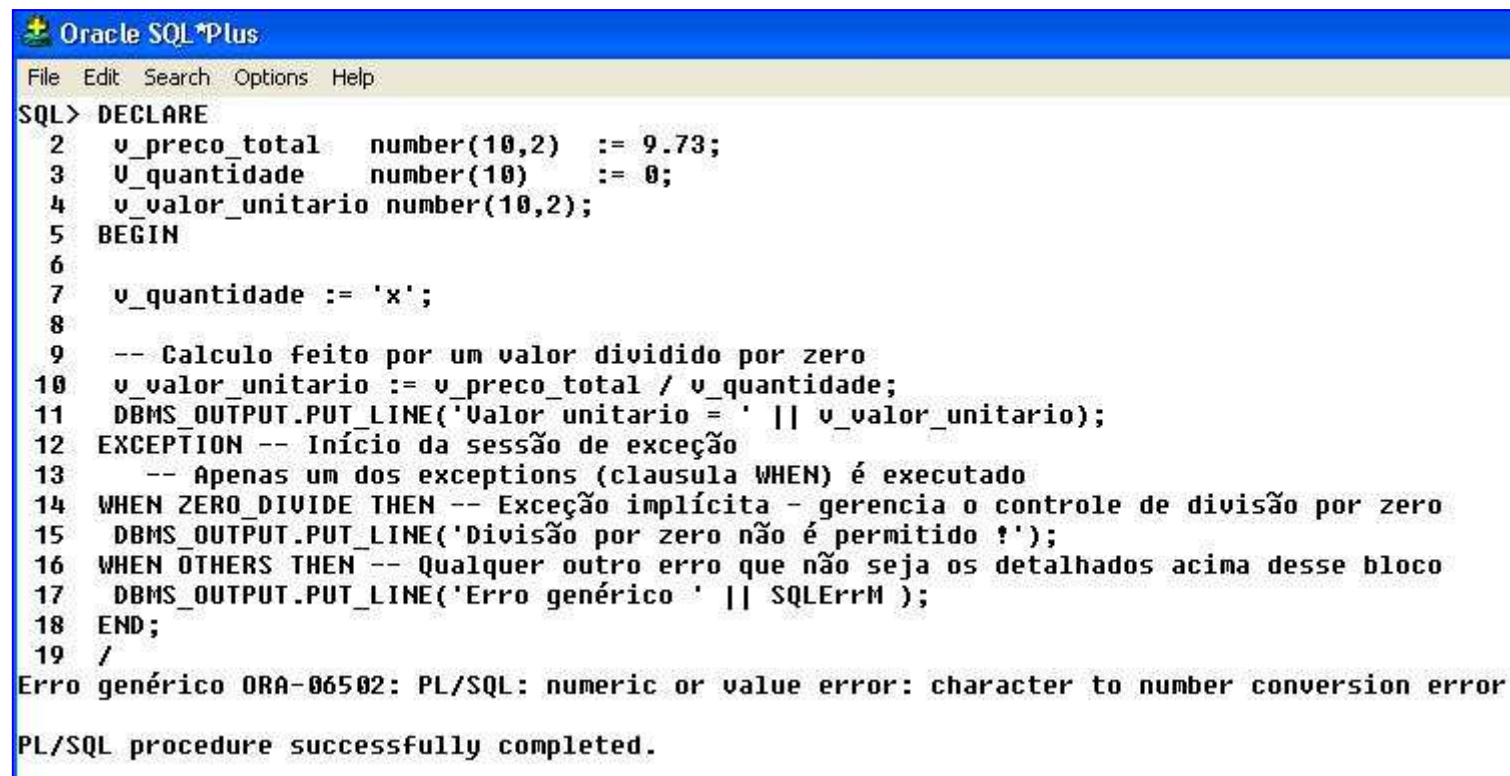
```
BEGIN
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    statement1;
    statement2;
  WHEN TOO_MANY_ROWS THEN
    statement1;
  WHEN OTHERS THEN
    statement1;
    statement2;
    statement3;
END;
```

Exemplo 1

```
Oracle SQL*Plus
File Edit Search Options Help
SQL> set serveroutput on
SQL> DECLARE
  2  v_preco_total    number(10,2)  := 9.73;
  3  v_quantidade     number(10)     := 0;
  4  v_valor_unitario number(10,2);
  5  BEGIN
  6    -- Calculo feito por um valor dividido por zero
  7    v_valor_unitario := v_preco_total / v_quantidade;
  8    DBMS_OUTPUT.PUT_LINE('Valor unitario = ' || v_valor_unitario);
  9  EXCEPTION -- Início da sessão de exceção
 10    -- Apenas um dos exceptions (clausula WHEN) é executado
 11  WHEN ZERO_DIVIDE THEN -- Exceção implícita - gerencia o controle de divisão por zero
 12    DBMS_OUTPUT.PUT_LINE('Divisão por zero não é permitido !');
 13  WHEN OTHERS THEN -- Qualquer outro erro que não seja os detalhados acima desse bloco
 14    DBMS_OUTPUT.PUT_LINE('Erro genérico ' || SQLERRM );
 15  END;
 16  /
Divisão por zero não é permitido !
PL/SQL procedure successfully completed.
```

Tratamento de Erros: Exception pré-definida

Exemplo 2



```
Oracle SQL*Plus
File Edit Search Options Help
SQL> DECLARE
  2  v_preco_total    number(10,2)  := 9.73;
  3  v_quantidade     number(10)     := 0;
  4  v_valor_unitario number(10,2);
  5  BEGIN
  6
  7    v_quantidade := 'x';
  8
  9    -- Calculo feito por um valor dividido por zero
 10    v_valor_unitario := v_preco_total / v_quantidade;
 11    DBMS_OUTPUT.PUT_LINE('Valor unitario = ' || v_valor_unitario);
 12  EXCEPTION -- Início da sessão de exceção
 13    -- Apenas um dos exceptions (clausula WHEN) é executado
 14  WHEN ZERO_DIVIDE THEN -- Exceção implícita - gerencia o controle de divisão por zero
 15    DBMS_OUTPUT.PUT_LINE('Divisão por zero não é permitido !');
 16  WHEN OTHERS THEN -- Qualquer outro erro que não seja os detalhados acima desse bloco
 17    DBMS_OUTPUT.PUT_LINE('Erro genérico ' || SQLERRM );
 18  END;
 19  /
Erro genérico ORA-06502: PL/SQL: numeric or value error: character to number conversion error
PL/SQL procedure successfully completed.
```

Tratamento de Erros: Exception pré-definida

Exemplo 3

```
Oracle SQL*Plus
File Edit Search Options Help
SQL> DECLARE
2  v_preco_total    number(10,2) := 9.73;
3  v_quantidade     number(10)   := 0;
4  v_valor_unitario number(10,2);
5  v_excecao_qtde_zero EXCEPTION;
6  BEGIN
7    if v_quantidade = 0 then
8      raise v_excecao_qtde_zero;
9    end if;
10   -- Calculo feito por um valor dividido por zero
11   v_valor_unitario := v_preco_total / v_quantidade;
12   DBMS_OUTPUT.PUT_LINE('Valor unitario = ' || v_valor_unitario);
13  EXCEPTION -- Início da sessão de exceção
14    -- Apenas um dos exceptions (clausula WHEN) é executado
15  WHEN ZERO_DIVIDE THEN -- Exceção implícita - gerencia o controle de divisão por zero
16    DBMS_OUTPUT.PUT_LINE('Divisão por zero não é permitido !');
17  WHEN v_excecao_qtde_zero THEN
18    DBMS_OUTPUT.PUT_LINE('É possível realizar o tratamento sem interromper o processamento');
19  WHEN OTHERS THEN -- Qualquer outro erro que não seja os detalhados acima desse bloco
20    DBMS_OUTPUT.PUT_LINE('Erro genérico ' || SQLERRM );
21  END;
22  /
É possível realizar o tratamento sem interromper o processamento
PL/SQL procedure successfully completed.
```

Tratamento de Erros: Exemplo 4

```
DECLARE
V_CD_GRUPO LOC_GRUPO.CD_GRUPO%TYPE := &COD_GRUPO;
V_DS_GRUPO LOC_GRUPO.DS_GRUPO%TYPE := '&NOM_GRUPO';
BEGIN
INSERT INTO LOC_GRUPO(CD_GRUPO, DS_GRUPO) VALUES
(V_CD_GRUPO, V_DS_GRUPO);
COMMIT;
EXCEPTION
WHEN DUP_VAL_ON_INDEX THEN --Exceção pré-definida pelo Oracle
    RAISE_APPLICATION_ERROR(-20001, 'Grupo já cadastrado');
WHEN OTHERS THEN --Exceção pré-definida pelo Oracle
    RAISE_APPLICATION_ERROR(-20002, 'Erro crítico - insert grupo veículo' ||
SQLERRM);
END;
```

Diretrizes para a Captura de Exceções

- **WHEN OTHERS** é a última cláusula.
- A palavra-chave **EXCEPTION** inicia a seção de tratamento de exceções.
- São permitidos vários handlers de exceção.
- Somente um handler é processado antes de se sair do bloco.

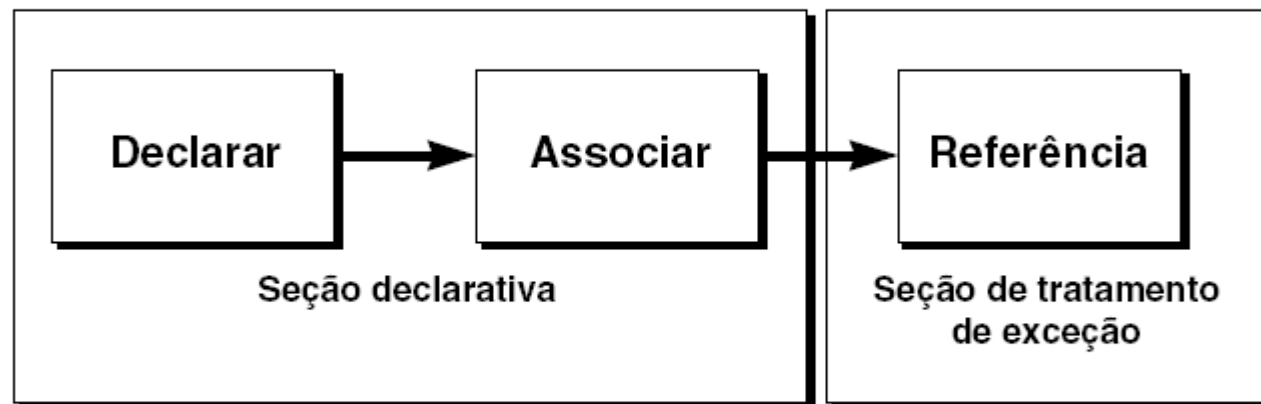
Tratamento de Erros: Exception

- *DUP_VAL_ON_INDEX(Ora-0001) - Ocorre na tentativa de armazenar um valor duplicado em uma coluna que possui chave-primária.*
- *INVALID_NUMBER(Ora-01722) - Ocorre na tentativa de converter uma string em número, caso a string não represente um valor numérico.*
- *NO_DATA_FOUND(Ora-1403) - Ocorre quando um comando SELECT.. INTO não retorna nenhuma linha.*
- *NO_DATA_FOUND(Ora-1403) - Ocorre quando um comando SELECT.. INTO não retorna nenhuma linha.*
- *TOO_MANY_ROWS(Ora-01422) - Ocorre quando um comando SELECT.. INTO retorna mais de um linha*
- *VALUE_ERROR(Ora-06502) - Ocorre quando houver um erro aritmético, de conversão, truncagem ou tamanho.*
- **OTHERS** - *Permite tratar outros erros, com a ajuda das funções SQLCODE e SQLERRM, que retorna o número do erro Oracle e o texto da mensagem de erro, respectivamente.*

Tratamento de Erros: RAISE_APPLICATION_ERROR

- *RAISE_APPLICATION_ERROR(cod_erro, 'msg');*
- *Fica armazenada na Package DBMS_STANDARD*
- *Permite a emissão de mensagens de erro definidas pelo usuário dentro de subprogramas*
- *Os erros podem ser relatados para a aplicação e evita-se o retorno de exceções não tratadas*
- *Cod_erro => valor inteiro negativo entre -20000 até -20999 (podendo se repetir)*

Capturando Erros Não Predefinidos do Oracle Server



- Nomeie a exceção
- Codifique o PRAGMA EXCEPTION_INIT
- Trate a exceção criada

Erro Não Predefinido

Capturar por número de erro do Oracle Server
-2292, uma violação de restrição de integridade.

```
DECLARE
  e_emps_remaining EXCEPTION;
  PRAGMA EXCEPTION_INIT (
    e_emps_remaining, -2292);
  v_deptno    dept.deptno%TYPE := &p_deptno;
BEGIN
  DELETE FROM dept
  WHERE      deptno = v_deptno;
  COMMIT;
EXCEPTION
  WHEN e_emps_remaining THEN
    DBMS_OUTPUT.PUT_LINE ('Cannot remove dept ' ||
      TO_CHAR(v_deptno) || '. Employees exist. ');
END;
```

1

2

3

```
CREATE TABLE errors (code NUMBER, message VARCHAR2(64), happened TIMESTAMP);
DECLARE
    name employees.last_name%TYPE;
    v_code NUMBER;
    v_errm VARCHAR2(64);
BEGIN
    SELECT last_name INTO name FROM employees WHERE employee_id = -1;
    EXCEPTION
        WHEN OTHERS THEN
            v_code := SQLCODE;
            v_errm := SUBSTR(SQLERRM, 1, 64);
            DEMS_OUTPUT.PUT_LINE('Error code ' || v_code || ': ' || v_errm);
-- Normally we would call another procedure, declared with PRAGMA
-- AUTONOMOUS_TRANSACTION, to insert information about errors.
            INSERT INTO errors VALUES (v_code, v_errm, SYSTIMESTAMP);
END;
```

- <http://www.ora-code.com>
- http://download-east.oracle.com/docs/cd/B12037_01/server.101/b10744/toc.htm.

Não basta saber, é preciso exercitar.

Lista de Exercícios

Copyright © 2013

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, dos Professor (autor).